数理解析研究所講究録
1021巻 1997年 96-111

The realization of the hybrid multi-modal logic theorem prover in the term rewriting system CafeOBJ

JAIST Yasuhito Suzuki(鈴木康人)

1. Introduction

A multi-modal logic can treat different modalities. In the recent research of computer science, these logics are tried to use to write actions of agents in a computer network. There are many researches about multi-modal logics, and here, we will put the M.Finger and D.Gabbay's result [FG94] basis. Compared with another reserach (for instance, [KW83]), we can say that the proof systems of [FG94] is hybrid. That means, the proof system of [KW83] can treat any combinations of different modal operators, but the proof system of [FG94] has some restrictions on order of modal operators; if modal logics L_1 and L_2 are given, and if we construct the multi-modal logic $L_1(L_2)$ by [FG94], then there are no modal operators of L_2 out of modal operators of L_1 at any formulas of $L_1(L_2)$. That is, the multi-modal logic $L_1(L_2)$ indicates that we see formulas of L_2 propositional variables of L_1. According to [FG94], this indicates a direct combination of global system L_1 and local system L_2.

CafeOBJ is a term rewriting system which is based on equational logic. Using CafeOBJ, this report aims to study a possibilty of realization of the method of [FG94], and gives concreate automated theorem prover for some modal logics (S4 and S5). We will adopt Beth tableau proof as the basis of the automated theorem prover. So, we suppose the reader already have the knowledge about Beth tableau proof system. Here, we just mention about theoretical results. The concreate program codes and expamples are represented in [S97].

First of all, we will introduce the S4- (and S5-) tableau proof system as a term rewriting system. We will mention that some useful properties to prove completeness theorem and decidablity of these system. Next, we will combine them to get hybrid multi-modal logic S4(S5) and S5(S4). We will refer the result of [FG94] related the combine method.

Here, we will not treat the correspondence between formal theories and programs of CafeOBJ. But this seems trivial by [CAFE].

2. S4

Here, we will introduce tableau proof system as term rewriting system. *Signed formula* is the formula which has a its truth value (either **T** or **F**) as prefix. Beth tableau consists of a set of signed formulas. S4 Beth tableau construction rules are as follows (see [F69]); suppose F is a sequence of signed modal formulas, $F_{\Box}$ is the sequence of signed modal formulas with

"$\mathbf{T}\ \Box$" as prefix in F and commas are the punctuation;

$$\frac{F, \mathbf{T}\ \Box\varphi}{F, \mathbf{T}\ \varphi}\ ,\ \frac{F, \mathbf{T}\ \varphi \wedge \psi}{F, \mathbf{T}\ \varphi, \mathbf{T}\ \psi}\ ,\ \frac{F, \mathbf{T}\ \neg\varphi}{F, \mathbf{F}\ \varphi}\ ,$$
$$\frac{F, \mathbf{F}\ \Box\varphi}{F_{\Box}, \mathbf{F}\ \varphi}\ ,\ \frac{F, \mathbf{F}\ \varphi \wedge \psi}{F, \mathbf{F}\ \varphi \mid F, \mathbf{F}\ \psi}\ ,\ \frac{F, \mathbf{F}\ \neg\varphi}{F, \mathbf{T}\ \varphi}\ .$$

Here, we think $\Diamond\varphi$, $\varphi \vee \psi$ and $\varphi \rightarrow \psi$ are the abbreviations of $\neg\Box\neg\varphi$, $\neg(\neg\varphi \wedge \neg\psi)$ and $\neg\varphi \vee \psi$, respectively. If we have a closed tableau of φ, then φ has a tableau proof. We denote this $\vdash_{S4} \varphi$. For instance, $\vdash_{S4} \Box p \rightarrow \Box\Box p$ as follows;

$$\cfrac{\mathbf{F}\ \neg(\Box p \wedge \neg\Box\Box p)}{\cfrac{\mathbf{T}\ \Box p \wedge \neg\Box\Box p}{\cfrac{\mathbf{T}\ \Box p, \mathbf{T}\ \neg\Box\Box p}{\cfrac{\mathbf{T}\ \Box p, \mathbf{F}\ \Box\Box p}{\cfrac{\mathbf{T}\ \Box p, \mathbf{F}\ \Box p}{\cfrac{\mathbf{T}\ \Box p, \mathbf{F}\ p}{\mathbf{T}\ p, \mathbf{F}\ p}}}}}}$$

But, in the above system, whether we can construct a closed tableau or not depends on an order of applying construction rules. For instance,

$$\cfrac{\mathbf{F}\ \neg(\Box p \wedge \neg\Box\Box p)}{\cfrac{\mathbf{T}\ \Box p \wedge \neg\Box\Box p}{\cfrac{\mathbf{T}\ \Box p, \mathbf{T}\ \neg\Box\Box p}{\cfrac{\mathbf{T}\ \Box p, \mathbf{F}\ \Box\Box p}{\cfrac{\mathbf{T}\ p, \mathbf{F}\ \Box\Box p}{\cfrac{\mathbf{F}\ \Box p}{\mathbf{F}\ p}}}}}}$$

is not closed. That is, we have to give construction rules where the order of rules is uniquely determined if the same formula is given, and always make a closed tableau if it exists.

Definition 1 *For any $i \in \{1, 2, 3, 4, 5\}$, let Q_i be a duplication free fi-*

nite sequents of signed formulas. We denote the empty sequence and the punctuation in Q_i *as* ϵ *and* $|$*, respectively. Both* $\mathsf{node}(Q_1, Q_2, Q_3, Q_4, Q_5)$ *and* $\mathsf{leaf}(Q_2)$ *are called node. We call the tree structure* $\mathcal{T}$*, that enjoys the following conditions, tableau tree;*

1. *any nodes of* $\mathcal{T}$ *are nodes,*
2. *any* $\mathsf{node}(Q_1, \ldots, Q_5)$ *has at most two children,*
3. *any* $\mathsf{leaf}(Q_2)$ *has no children,*
4. *the root of* $\mathcal{T}$ *is in the form* $\mathsf{node}(\epsilon, \epsilon, \epsilon, \epsilon, Q_5)$.

Definition 2 *Let a node of a tableau tree* $N(= \mathsf{node}(Q_1, \ldots, Q_5))$ *be given. The followings are tableau construction rule. Applying rule to* N*, we consider that a new node* $N'^{(\prime)}$ *are added to given tabuleau tree. In each rule, we will express modified sequence of new node(s)* $N'^{(\prime)}$ $(= \mathsf{node}(Q_1'^{(\prime)}, \ldots, Q_5'^{(\prime)}))$*, thus if there are no references of sequences in a rule, they are same to the parents' sequences. We apply a rule by the depth-first search. Suppose* $\mathrm{cls}(Q)$ *means sequence* Q *has a pair of signed propositional variable* $\mathbf{T}\ p$ *and* $\mathbf{F}\ p$*, and* $\mathrm{loop}(N)$ *means there exists the ancestor of* N*, which is same to* N*. And, let* Q *is duplication free finite (or empty) sequence of signed formulas,c is a propositional variable and* $tf \in \{\mathbf{T}, \mathbf{F}\}$.

1. $Q_5 = tfc|Q \Rightarrow \begin{cases} Q_5' = Q \\ Q_2' = tfc|Q_2 \end{cases}$

2. $Q_5 = \mathbf{T}\ (\varphi \wedge \psi)|Q \Rightarrow Q_5' = \mathbf{T}\ \varphi|\mathbf{T}\ \psi|Q$

3. $Q_5 = \mathbf{T}\ \neg\varphi|Q \Rightarrow Q_5' = \mathbf{F}\ \varphi|Q$

4. $Q_5 = \mathbf{T}\ \Box\varphi|Q \Rightarrow \begin{cases} Q_1' = \mathbf{T}\ \Box\varphi|Q_1 \\ Q_5' = \mathbf{T}\varphi|Q \end{cases}$

5. $Q_5 = \mathbf{F}(\varphi \wedge \psi)|Q \Rightarrow \begin{cases} Q_3' = \mathbf{F}(\varphi \wedge \psi)|Q_3 \\ Q_5' = Q \end{cases}$

6. $Q_5 = \mathbf{F}\neg\varphi|Q \Rightarrow Q_5' = \mathbf{T}\varphi|Q$

7. $Q_5 = \mathbf{F}\Box\varphi|Q \Rightarrow \begin{cases} Q_4' = \mathbf{F}\varphi|Q_4 \\ Q_5' = Q \end{cases}$

8. $\left.\begin{array}{l} \textit{not}\ \mathrm{cls}(Q_2) \\ Q_3 = \mathbf{F}(\varphi \wedge \psi)|Q \\ Q_5 = \epsilon \end{array}\right\} \Rightarrow \begin{cases} Q_3' = Q_3'' \\ Q_5' = \mathbf{F}\varphi|\epsilon \\ Q_5'' = \mathbf{F}\psi|\epsilon \end{cases}$

9. $\left.\begin{array}{l} \textit{not}\ \mathrm{cls}(Q_2) \\ \textit{not}\ \mathrm{loop}(N) \\ Q_4 = \mathbf{F}\varphi|\epsilon \\ Q_3 = Q_5 = \epsilon \end{array}\right\} \Rightarrow \begin{cases} Q_1' = Q_2' = Q_3' = Q_4' = \epsilon \\ Q_5' = \mathbf{F}\varphi|Q_1 \end{cases}$

10. $\left.\begin{array}{l} \textit{not}\ \mathrm{cls}(Q_2) \\ \textit{not}\ \mathrm{loop}(N) \\ Q_4 = \mathbf{F}\varphi|Q \\ Q \neq \epsilon \\ Q_3 = Q_5 = \epsilon \end{array}\right\} \Rightarrow \begin{cases} Q_4' = Q \\ Q_4'' = \mathbf{F}\varphi|\epsilon \end{cases}$

$$\textit{11.}\quad \left.\begin{array}{l} \text{not cls}(Q_2) \\ \\ Q_3 = Q_4 = Q_5 = \epsilon \end{array}\right\} \Rightarrow \mathsf{leaf}(Q_2)$$

$$\textit{12.}\quad \left.\begin{array}{l} \text{cls}(Q_2) \\ \\ Q_5 = \epsilon \end{array}\right\} \Rightarrow \mathsf{leaf}(Q_2)$$

$$\textit{13.}\quad \left.\begin{array}{l} \textit{not}\ \text{cls}(Q_2) \\ \\ \text{loop}(N) \\ \\ Q_4 \neq \epsilon \\ \\ Q_3 = Q_5 = \epsilon \end{array}\right\} \Rightarrow \mathsf{leaf}(Q_2)$$

Definition 3 $N(= \mathsf{node}(Q_1, Q_2, Q_3, Q_4, Q_5))$ *is closed if*

- $\text{cls}(Q_2)$, *or*
- *N has at least one closed child N' by applying rule 10,*
- *all children of N are closed.shi*

For any given formula φ, if we can construct the tableau which the root in the form $\mathsf{node}(\epsilon, \epsilon, \epsilon, \epsilon, \mathbf{F}\varphi|\epsilon)$ is closed, we denote this as $\vdash_{\mathrm{cS4}} \varphi$. About the above rules, we can say that the following facts hold.

Lemma 4 *In any tableau construction, there exists the common order of applying rules, as follows;*

1st step: Apply the one of rules from rule 1 to rule 7, until $Q_5 = \epsilon$.

2nd step: *If* $Q_3 \neq \epsilon$ *and not* $\mathrm{cls}(Q_2)$*, then go to 1st step after the applying rule 8.*

3rd step: *If we can apply rule 10, then apply it.*

4th step: *If we can apply rule 9, then go to 1st step after the applying rule 9.*

end step: *If we can apply the one of rule 11, rule 12 and rule 13, then apply it and close the making of a branch.*

Lemma 5 *Our tableau construction cannot make an infinite depth branch.*

Lemma 6 *Any formulas in any node* N *are applied extension rule at least one time in the descendant of* N*.*

Proposition 7 *For any given formula* φ*, our tableau construction always terminates.*

Defining a peculiar notion of realizability, we can prove the soundness and completeness of our tableau proof system. Here, we just mention this realizable notion and the soundness and completeness. About Kripke model, for instance, see [CZ97].

Definition 8 *Suppose* $\mathcal{M}$ *is a Kripke model of S4, and* x *and* y *are possible worlds of* $\mathcal{M}$*. A finite set of signed formulas* $U = (\{\mathbf{T}\varphi_1, \ldots, \mathbf{T}\varphi_n, \mathbf{F}\psi_1, \ldots, \mathbf{F}\psi_m\})$ *is realizable if there exists* $\mathcal{M}$ *and* x *such that*

$$(\mathcal{M}, x) \models \varphi_1, \ldots, (\mathcal{M}, x) \models \varphi_n,$$

$$(\mathcal{M}, x) \not\models \psi_1, \ldots, (\mathcal{M}, x) \not\models \psi_m.$$

Denote this as $\mathrm{rl}(\mathcal{M}, U, x)$. $\mathsf{node}(Q_1, Q_2, Q_3, Q_4, Q_5)$ *is realizable if there exists* $\mathcal{M}$, x *and* y *such that* xRy, $\mathrm{rl}(\mathcal{M}, U, x)$ *and* $\mathrm{rl}(\mathcal{M}, U', y)$ *where*

$$\begin{aligned} U &= Q_2 \cup Q_3 \cup Q_5 \cup \{\mathbf{T}(p \vee \neg p)\}, \\ U' &= Q_1 \cup Q_4 \cup \{\mathbf{T}(p \vee \neg p)\}. \end{aligned}$$

Using tableau extension rule, wa can say that realizable node has at least one realizable child. And,

Lemma 9 *A node is relaizable iff it is not closed.*

Theorem 10 $\vdash_{\mathrm{cS4}} \varphi$ *iff* $\models \varphi$.

3. S5

For instance, S5 Beth tableau is introduced in [F77], but the method in [F77] is not appropriate to constructing term rewriting system. By corresponding each node to the tableau in [F77], we can construct S5 tableau system as term rewriting system.

From now on, we denote a duplication free finite sequence of signed formulas, the empty sequence of signed formula and punctuation symbol in sequence of signed formulas as Q_i, ϵ and $|$, respectively. And, we denote a duplication free finite sequence of pairs of sequences of signed formulas, the empty sequence of sequences of pairs and punctuation as P_j, E and $\ddagger$. Then,

Definition 11 node(P_1, P_2, P_3) *is called (S5-) node. We call $\mathcal{T}$ tableau tree if the following conditions are satisfied;*

1. *$\mathcal{T}$ is tree of S5-node,*
2. *Each S5-node has at most two chilrdren,*
3. *a root of $\mathcal{T}$ is in the form* node$((\epsilon, \epsilon) \ddagger E, E, P_3)$

Definition 12 *Suppose a node $N(=$ node$(P_1, P_2, P_3))$ is given. Via the following extension rules, if we can get a child of N, we express this as $N'^{(\prime)}(=$ node$(P_1'^{(\prime)}, P_2'^{(\prime)}, P_3'^{(\prime)}))$. As similar to the case of S4, we just mention about modified factors in the children of N in the following rules;*

1. node$((Q, \epsilon) \ddagger E, P, E)$ *has no children.*
2. $P_3 = (\epsilon, \epsilon) \ddagger P \Rightarrow P_3' = P$
3. $P_3 = (Q_1, tf\ c|Q_2) \ddagger P \Rightarrow P_3' = (tf\ c|Q_1, Q_2) \ddagger P$
4. $P_3 = (Q_1, \mathbf{T}\neg\varphi|Q_2) \ddagger P \Rightarrow P_3'(Q_1, \mathbf{F}\varphi|Q_2) \ddagger P$
5. $P_3 = (Q_1, \mathbf{F}\neg\varphi|Q_2) \ddagger P \Rightarrow P_3'(Q_1, \mathbf{T}\varphi|Q_2) \ddagger P$
6. $P_3 = (Q_1, \mathbf{T}\varphi \wedge \psi|Q_2) \ddagger P \Rightarrow P_3' = (Q_1, \mathbf{T}\varphi|\mathbf{T}\psi|Q_2) \ddagger P$
7. $P_3 = (Q_1, \mathbf{F}\varphi \wedge \psi|Q_2) \ddagger P \Rightarrow \begin{cases} P_3' = (Q_1, \mathbf{F}\varphi|Q_2) \ddagger P \\ P_3'' = (Q_1, \mathbf{F}\psi|Q_2) \ddagger P \end{cases}$
8. $\left.\begin{array}{l} P_3 = (Q_3, \mathbf{T}\Box\varphi|Q_4) \ddagger P_5 \\ P_1 = (Q_1, Q_2) \ddagger P_4 \end{array}\right\} \Rightarrow \begin{cases} P_3' = (Q_3, Q_4) \ddagger P_5 \\ P_1' = (Q_1, \mathbf{T}\varphi|Q_2) \ddagger P_4 \end{cases}$

9. $P_3 = (Q_1, \mathbf{F}\Box\varphi|Q_2) \ddagger P \Rightarrow P_3' = (\epsilon, \mathbf{F}\varphi|\epsilon) \ddagger (Q_1, Q_2) \ddagger P$

10. $$\left.\begin{array}{l} P_3 = (Q_1, \epsilon) \ddagger P \\ P_2 = P' \end{array}\right\} \Rightarrow \left\{\begin{array}{l} P_3' = P \\ P_2' = (Q_1, \epsilon) \ddagger P' \end{array}\right.$$

11. $$\left.\begin{array}{l} P_3 = E \\ P_1 = (Q_1, tf\ c|Q_2) \ddagger P \end{array}\right\} \Rightarrow \left\{\begin{array}{l} P_3' = E \\ P_1' = (tf\ c|Q_1, Q_2) \ddagger P \end{array}\right.$$

12. $$\left.\begin{array}{l} P_3 = E \\ P_1 = (Q_1, \mathbf{T}\neg\varphi|Q_2) \ddagger P \end{array}\right\} \Rightarrow \left\{\begin{array}{l} P_3' = E \\ P_1' = (Q_1, \mathbf{F}\varphi|Q_2) \ddagger P \end{array}\right.$$

13. $$\left.\begin{array}{l} P_3 = E \\ P_1 = (Q_1, \mathbf{F}\neg\varphi|Q_2) \ddagger P \end{array}\right\} \Rightarrow \left\{\begin{array}{l} P_3' = E \\ P_1' = (Q_1, \mathbf{T}\varphi|Q_2) \ddagger P \end{array}\right.$$

14. $$\left.\begin{array}{l} P_3 = E \\ P_1 = (Q_1, \mathbf{T}\varphi \wedge \psi|Q_2) \ddagger P \end{array}\right\} \Rightarrow \left\{\begin{array}{l} P_3' = E \\ P_1' = (Q_1, \mathbf{T}\varphi|\mathbf{T}\psi|Q_2) \ddagger P \end{array}\right.$$

15. $$\left.\begin{array}{l} P_3 = E \\ P_1 = (Q_1, \mathbf{F}\varphi \wedge \psi|Q_2) \ddagger P \end{array}\right\} \Rightarrow \left\{\begin{array}{l} P_3' = P_3'' = E \\ P_1' = (Q_1, \mathbf{F}\varphi|Q_2) \ddagger P \\ P_1'' = (Q_1, \mathbf{F}\psi|Q_2) \ddagger P \end{array}\right.$$

16. $$\left.\begin{array}{l} P_3 = E \\ P_1 = (Q_1, \mathbf{T}\Box\varphi|Q_2) \ddagger P \end{array}\right\} \Rightarrow \left\{\begin{array}{l} P_3' = E \\ P_1' = (Q_1, \mathbf{T}\varphi|Q_2) \ddagger P \end{array}\right.$$

17. $$\left.\begin{array}{l} P_3 = E \\ P_1 = (Q_1, \mathbf{F}\Box\varphi|Q_2) \ddagger P \end{array}\right\} \Rightarrow \left\{\begin{array}{l} P_3' = E \\ P_1' = (Q_1, Q_2) \ddagger (\epsilon, \mathbf{F}\varphi|\epsilon) \ddagger P \end{array}\right.$$

18. $$\left.\begin{array}{l} P_3 = E \\ P_1 = (Q_1, \epsilon) \ddagger (Q_2, Q_3) \ddagger P \end{array}\right\} \Rightarrow \left\{\begin{array}{l} P_3' = (Q_2, Q_3) \ddagger P \\ P_1' = (Q_1, \epsilon) \ddagger E \end{array}\right.$$

You can easily notice that the number of logical connectives ($\wedge$, $\neg$ and $\Box$) of a child node k is same to the number of the parent node via the rule 1, 2, 3, 10, 11 and 18, and $k+1$ is same to the number of the parent node via the another rules. As similar for the case S4, we can find there exists some peculiar order of applying rules. That is,

Lemma 13 *In any tableau construction, there exists the following common order of applying rules;*

1st step:	*Apply one rule between rule 3 and rule 9 until $P_3 = (Q_1, \epsilon) \ddagger P$ holds.*
2nd step:	*Apply either rule 2 or rule 10.*
3rd step:	*Go to 1st step if $P_3 \neq E$.*
4th step:	*Apply one rule between rule 11 and rule 17 until $P_1 = (Q_1, \epsilon) \ddagger P$ holds.*
5th step:	*Apply rule 18 and go to 1st step if $P_1 \neq (Q_1, \epsilon) \ddagger E$.*
6th step:	*Apply rule 1.*

Proposition 14 *Our tableau construction always terminate.*

The beginning of this section, we said that each node in our tableau corresponds to the S5 tableau in [F77]. The correct meaning of this sentence is introduced via the following definition about realizable notion.

Definition 15 *Suppose* $\mathsf{node}(P_1, P_2, P_3)$ *is given, where the number of sequences except* (ϵ, ϵ) *of* P_i *is* a_i, *and* $s(P_i) = \{tf\varphi : tf\varphi \text{ appears in } P_i\}$.

And we think that P_i has $(Q_1^i, Q_2^i)_1$, ..., $(Q_1^i, Q_2^i)_{a_i}$ elements where each $(Q_1^i, Q_2^i)_j \neq (\epsilon, \epsilon)$, and $s((Q_1^i, Q_2^i)_j) = s((Q_1^i, Q_2^i)_j \ddagger E) = \{\mathbf{T}\varphi_{1j}^i, \ldots, \mathbf{T}\varphi_{b_{ij}j}^i, \mathbf{F}\psi_{1j}^i, \ldots, \mathbf{F}\psi_{c_{ij}j}^i\}$. $\mathsf{node}(P_1, P_2, P_3)$ is realizable if there exists the S5-model $\mathcal{M}(= \langle W, R, V\rangle)$ such that $W = \{x_1^1, \ldots, x_{a_1}^1, x_1^2, \ldots, x_{a_2}^2, x_1^3, \ldots, x_{a_3}^3\}$ and for any i and j,

$$(\mathcal{M}, x_j^i) \models \varphi_{11}^1, \ldots, (\mathcal{M}, x_j^i) \models \varphi_{b_{11}1}^1$$

$$(\mathcal{M}, x_j^i) \not\models \psi_{11}^1, \ldots, (\mathcal{M}, x_j^i) \not\models \psi_{c_{11}1}^1$$

$$(\mathcal{M}, x_j^i) \models \varphi_{1j}^i, \ldots, (\mathcal{M}, x_j^i) \models \varphi_{b_{ij}j}^i$$

$$(\mathcal{M}, x_j^i) \not\models \psi_{1j}^i, \ldots, (\mathcal{M}, x_j^i) \not\models \psi_{c_{ij}j}^i$$

This definition asserts that the **T**-signed formulas in the top of P_1 hold on any points in $\mathcal{M}$, and **F**-signed formulas in the top of P_1 does not hold on any points in $\mathcal{M}$. Because, the next proposition holds on any S5-model.

Proposition 16 *(proposition 3.7 in [CZ97]) Suppose x is a point in a model $\mathcal{M}$ built on a transitive frame and φ an arbitrary formula. Then for every $y \in C(x)$,*

$$(\mathcal{M}, x) \models \Box\varphi \;\; \textit{iff} \;\; (\mathcal{M}, y) \models \Box\varphi,$$

$$(\mathcal{M}, x) \models \Diamond\varphi \;\; \textit{iff} \;\; (\mathcal{M}, y) \models \Diamond\varphi.$$

$C(x)$ $(= \{y : xRy \text{ and } yRx\})$ is a cluster of x, and W of every S5-model can be represented into one cluster. Hence, we will define closed notion, as follows;

Definition 17 *A node N is closed if*

- *there exists a pair of signed propositional variables* $\mathbf{T}p$ *and* $\mathbf{F}p$ *in* Q_1^1 *and* Q_i^2 *for some* i, *if* $N = \mathsf{node}((Q_1^1, \epsilon) \ddagger E, (Q_1^2, \epsilon) \ddagger \ldots \ddagger (Q_n^2, \epsilon) \ddagger E, E)$, *or*
- *all chilrdern are closed.*

As similar to the case S4, for any formula φ, if we can construct the tableau tree which root $\mathsf{node}((\epsilon, \epsilon) \ddagger E, E, (\epsilon, \mathbf{F}\varphi|\epsilon) \ddagger E)$ is closed, then we write $\vdash_{\mathrm{cS5}} \varphi$. And,

Lemma 18 *A node is relaizable iff it is not closed.*

Theorem 19 $\vdash_{\mathrm{cS5}} \varphi$ *iff* $\models \varphi$.

4. Hybrid multi-modal logic theorem prover

Here, we will define a multi-modal logic theorem prover. In [FG94], M.Finger and D.Gabbay adopt until $\mathcal{U}$ and since $\mathcal{S}$ operators as temporal (modal) operator. Because they treat temporal logic, and these two operators are base rather than $\Box$. And $\Box\varphi$ is introduced as the abbreviation of using $\mathcal{U}$ and $\mathcal{S}$. But, some minor changes produces that same results in [FG94] are holds on the our system. M.Finger and D.Gabbay call their logic temporalized logic, but we call our logic Hybrid multi-modal logic to destinguish the former.

Definition 20 *Suppose propositional modal languages* L_1 *and* L_2 *are given,* *and* $\wedge$, $\neg$ *and* $\Box$ *are logical connectives of* L_1. *Then, we define the hybrid language of* $L_1(L_2)$ *as follows;*

1. $L_2 \subseteq L_1(L_2)$,
2. $\varphi \in L_1(L_2)$ *implies* $\Box\varphi, \neg\varphi \in L_1(L_2)$,
3. $\varphi, \psi \in L_1(L_2)$ *implies* $(\varphi \wedge \psi) \in L_1(L_2)$.

For the satisfiability of $L_1(L_2)$, as follows;

Definition 21 *Suppose* $\mathcal{M}_2$ *and* x_2 *are model of* L_2 *and the root of* $\mathcal{M}_2$. *We denote* $(\mathcal{M}_2, x_2) \models \varphi$ *and* p *as* $\mathcal{M}_2 \models_2 \varphi$ *and* L_2*-formula, respectively. Let a frame* $F = \langle W, R\rangle$ *of* L_1 *be given, and* v *is a function* $W \to \mathcal{CM}_2$ *where* $\mathcal{CM}_2$ *is the class of models of* L_2. *Then, for any* $x \in W$,

$$
\begin{aligned}
(\langle F, v\rangle, x) \models p \quad & \textit{iff} \quad \mathcal{M}_2 \models_2 p \textit{ if } v(x) = \mathcal{M}_2,\\
(\langle F, v\rangle, x) \models \neg\varphi \quad & \textit{iff} \quad (\langle F, v\rangle, x) \not\models \varphi,\\
(\langle F, v\rangle, x) \models \varphi \wedge \psi \quad & \textit{iff} \quad (\langle F, v\rangle, x) \models \varphi \textit{ and } (\langle F, v\rangle, x) \models \psi,\\
(\langle F, v\rangle, x) \models \Box\varphi \quad & \textit{iff} \quad \textit{for all } y\ xRy \textit{ implies}((\langle F, v\rangle, y) \models \varphi,\\
\langle F, v\rangle \models \varphi \quad & \textit{iff} \quad (\langle F, v\rangle, x) \models \varphi \textit{ for all } x \in W,\\
F \models \varphi \quad & \textit{iff} \quad \langle F, v\rangle \models \varphi \textit{ for all } v\ ,\\
C \models \varphi \quad & \textit{iff} \quad F \models \varphi \textit{ for all } F \in C.
\end{aligned}
$$

If we know what a class of frames C *corresponds to given logic, sometimes we write* $\models \varphi$ *instead of* $C \models \varphi$.

As for the derivability $\vdash$ is able to consider many different definition. Here, we consider about $\vdash_{\mathrm{cS4}}$ and $\vdash_{\mathrm{cS5}}$.

Definition 22 *Suppose φ is a S4(S5)- (S5(S4)-) fomrula. We extend the defintion of a closed node, as follows; suppose χ, $\chi_1, \ldots, \chi_m$ are S5- (S4-) subformulas of φ then, node N is also closed if*

- *χ is assigned the sign* **T** *in N in spite of $\vdash_{\mathrm{cS5(cS4)}} \neg\chi$ holds*
- *χ is assigned the sign* **F** *in N in spite of $\vdash_{\mathrm{cS5(cS4)}} \chi$ holds.*
- *$\chi_1, \ldots, \chi_n$ have the sign* **T**, *and $\chi_{n+1}, \ldots, \chi_m$ have the sign* **F** *in spite of*

$$\vdash_{\mathrm{cS5(cS4)}} \neg(\chi_1 \wedge \ldots \wedge \chi_n \wedge (\neg\chi_{n+1} \wedge \ldots \wedge \neg\chi_m))$$

For any given S4(S5)- (S5(S4)-) formula φ, if we can construct the root of the tableau of φ is closed, then we write $\vdash_{\mathrm{cS4(S5)}} \varphi(\vdash_{\mathrm{cS5(S4)}} \varphi)$.

Then, clearly, we can say the following theorem,

Theorem 23 $\vdash_{\mathrm{cS4(S5)}} \varphi$ *iff* $\models \varphi$.

Theorem 24 $\vdash_{\mathrm{cS5(S4)}} \varphi$ *iff* $\models \varphi$.

Theorem 25 *Both $\vdash_{\mathrm{cS4(S5)}}$ and $\vdash_{\mathrm{cS5(S4)}}$ are decidable.*

参考文献

[CAFE] `http://ldl-www.jaist.ac.jp:8080/`

[CZ97] A.Chagrov and M.Zakharyaschev, *Modal logic*, Oxford, 1997.

[F69] M.Fitting, *Intuitionistic logic model theory and forcing*, North-Holland, 1969.

[F77] M.Fitting, A tableau system for propositional S5, In the *Notre Dame Journal of Formal Logic, volume XVIII, number 2*, pp 292-295, 1977.

[FG94] M.Finger and D.Gabbay, Chapter 14, Adding a temporal dimension to a logic system, In *Temporal logic : Mahtematical foundations and computational aspects, volume 1*, Oxford, pp523-562, 1994.

[KW83] M.Kracht and F.Wolter, Properties of independently axiomatizable bimodal logics, in the *Journal of Symbokic Logic, volume 56, number 4, pp 1469-1485*, 1983.

[S97] Y.Suzuki, *The realization of hybrid multi-modal logic theorem prover in the term rewriting system CafeOBJ*, to be appear in Technical Memorandom in JAIST, 1997.